

TAMPEREEN AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka

Tutkintotyö

Micael Laine

DIGI-TV:N MHP-SOVELLUSSTANDARDIN MUKAINEN SOVELLUSKEHITYS

Työn ohjaaja: Pekka Pöyry
Työn teettäjä: LähiTV-hanke, valvojana Pekka Pöyry
Tampere 2005

TAMPEREEN AMMATTIKORKEAKOULU

Tietotekniikka

Ohjelmistotekniikka

Laine, Micael Digi-TV:n MHP-sovellusstandardin mukainen sovelluskehitys

Tutkintotyö 34 sivua

Työn ohjaaja Pekka Pöyry

Työn teettäjä LähiTV-hanke, valvojana Pekka Pöyry

Toukokuu 2005

Hakusanat digi-tv, MHP, DVB

TIIVISTELMÄ

Siirtyminen digitaaliseen televisioon on koko television historian suurin muutos. Yksi sen tärkeimmistä ominaisuuksista on vuorovaikuteisuus käyttäjän ja palveluntarjoajan välillä. Tämän vuorovaikuteisuuden mahdollistaa DVB-organisaation kehittämä MHP-tekniikka.

Tässä insinööriytyössä on tutustuttu MHP-sovellusstandardiin ja osallistuttu MHP-standardin mukaisen digi-tv-sovelluksen luomiseen. Sovelluksen ideana on antaa palautetta esitettävästä TV-ohjelmasta. Sovellus luo esitettävän ohjelman perusteella valikon, josta käyttäjä voi valita haluamansa äänestysvaihtoehdon. Käyttäjä voi myös katsella äänestystilastoja omasta televisiostaan.

Sovellus käyttää digiboxin ethernet-paluukanavaa käyttäjän palautteen lähettämiseen palvelimelle. Annetut äänet tallennetaan tietokantaan. Käyttäjien palautteista koostetaan tilastoja, joita voidaan myöhemmin hyödyntää. Toteutustekniikkana on käytetty MHP-ohjelmointia (Javan Xlet-tekniikka) ja palvelinohjelmointia, joka on toteutettu Apache- ja MySQL tekniikoilla. Sovelluskehitysympäristönä on käytetty OpenMHP-alustaa, johon kuuluu emulaattori.

Työssä on keskitytty MHP-sovelluksen tekniseen toteutukseen, ei niinkään ulkoasuun tai käytettävyyteen. Tulevaisuudessa sovellus on LähiTV-hankkeen käytössä ja sen kehitystä jatketaan Tampereen AMK:ssa.

TAMPERE POLYTECHNIC

Computer Systems Engineering

Software Engineering

Laine, Micael Software development using MHP -standard

Engineering Thesis 34 pages

Thesis Supervisor Pekka Pöyry

Commissioning company: LähiTV-hanke. Supervisor: Pekka Pöyry

May 2005

Keywords: ITV, DTV, MHP, DVB

ABSTRACT

The digital transition is the biggest change in the television medium since the advent of television itself. DTV technology provides a host of new opportunities for public television to, among other things, provide interactive programs never before possible with today's analog broadcasting standard. The interactivity is made possible by the DVB-MHP technique. MHP (Multimedia Home Platform) is the open middleware system designed by the DVB project and it defines a generic interface between interactive digital applications and the terminals on which those applications execute.

This thesis focuses to the MHP standard and participate MHP software development. The main idea of created application is to give feedback to the TV-program presented. The application creates a menu based on the program, from which the wanted option can be chosen. The voting results can also be viewed from the users own TV-set. When sending the feedback to the server, the application uses the Ethernet return channel. The given votes are saved into the database. The used development environment is the OpenMHP platform, including emulator.

The main focus in this thesis is on technical implementation, layout and usability are left with less attention. In the future created application will be used in the Neighbourhood TV project (LähiTV-hanke) and its development will be continued in Tampere polytechnic.

SISÄLLYSLUETTELO

TIIVISTELMÄ

ABSTRACT

SISÄLLYSLUETTELO	1
LYHENTEIDEN JA MERKKIEN SELITYKSET	3
1 JOHDANTO	5
2 DIGITAALINEN TELEVISIO	6
2.1 Historia	6
2.2 Digitaaliset lisäarvopalvelut	6
2.3 Paluukanava	7
2.4 DVB-standardi	7
2.5 Datavirta	8
3 MHP-SOVELLUSSTANDARDI	9
3.1 MHP-standardin historia	9
3.2 Toimintaperiaate	9
3.3 MHP -profiilit	9
3.4 Arkkitehtuuri	10
3.5 Syötteet	11
3.6 Katsojan tunnistaminen	12
3.7 Muistivaatimus	12
3.8 Turvallisuus	12
4 MHP-SOVELLUS	13
4.1 MHP-sovelluksen elinkaari	13
4.2 Sisältoformaatit	13
4.3 MPEG-2-standardi	14
4.4 Data- ja objektikaruselli	15
4.5 Tiedostojen käsittely	16
4.6 Syötteiden käsittely	16
4.7 Sovellustyyppit	17
4.8 JavaTV API:n kategoriat	18
4.9 Xlet-rajapinta	18
4.10 Grafiikka	20
4.11 Ääni	21
5 Äänestyssovellus	22
5.1 Sovellusympäristö	22
5.2 OpenMHP	22
5.3 Asiakas-palvelin -tiedonsiirto	23
5.4 Järjestelmän toiminnot	24
5.4.1 Perustila	24
5.4.2 Äänestys	24
5.4.3 Annettujen äänien tulostus	24
5.5 Ohjelmistoarkkitehtuuri, moduulit ja prosessit	25
5.6 Virheidennkäsittely	27
5.7 Testaus	27

6	Yhteenveto.....	29
	LÄHTEET	31

LYHENTEIDEN JA MERKKIEN SELITYKSET

API Application Programming Interface; Käyttöjärjestelmän tai sovelluksen tarjoama rajapinta, jonka kautta niihin voidaan tehdä ohjelmakutsuja ulkopuolelta.

AWT Abstract Windowing Toolkit; Kokoelma Java-kielisiä rajapintoja (ks. API), jotka tarjoavat valmiita komponentteja käyttöliittymän ohjelmointiin. Uusimpien Javan versioiden mukaisissa sovelluksissa käytetään AWT:stä kehittyneempää versiota, Swingiä.

Bluetooth; Lyhyen kantaman radiotaajuudella toimiva yhteystapa, jolla matkapuhelimet, tietokoneet ja PDA-laitteet voivat kommunikoida keskenään langattomasti.

Broadcaster; televisiolähetystoiminnan harjoittaja.

CAT Conditional Acces Table; Sisältää tiedot salauksen purkuun.

DSM-CC Digital Storage Media-Command and Control; Tekniikka, joka mahdollistaa sovellustiedostojen lähettämisen osana DVB-lähetystä.

DVB Digital Video Broadcast; Digitaalisten televisiolähetysten tekninen standardi.

DVB-SI Digital Video Broadcast Service Information; Palveluinformaatiojärjestelmä.

ELG - European Launching Group; Digitaalisen television kehittämistä valvova elin.

GUI Graphical User Interface; Tietokoneen graafinen käyttöliittymä.

GPRS General Packet Radio Services; Teknologia, joka mahdollistaa matkapuhelinten pakettimuotoisen langattoman dataliikenteen.

HTTP Hypertext Transfer Protocol; Sääntömäärittely tiedostojen lähettämiseen webissä.

IP Internet Protocol; Protokolla, joka määrittää säännöt informaation lähettämiseksi tietokoneelta toiselle Internetissä. Jokaisella protokollaa käyttävällä laitteella on oltava IP-osoite, jonka perusteella kone löytyy verkossa.

Java; Ohjelmointikieli, joka on suunniteltu erityisesti Internetin tietoliikennepäristöjä ajatellen. Java on olioohjelmointikieli, mikä tarkoittaa että sillä laadittu sovellus koostuu luokista ja niiden välisistä suhteista.

JavaTV API; Rajapintakokoelma, jolla voidaan kontrolloida kodin multtimediapäätelaitteiden televisio toimintoja Java-sovelluksesta käsin. API on otettu osaksi DVB-MHP-määritystä.

JMF Java Multimedia Framework; Java-kielen sovelluskehys, joka mahdollistaa audion, videon ja muiden aikariippuvaisten medioiden lisäämisen Java-sovelluksiin.

MHP Multimedia Home Platform; DVB-projektin laatima standardi digitaalisen television vastaanottimien väliohjelmistolle (middleware).

Multipleksi / Muksi; Digitaalisen television lähetyksessä käytettävä bittivirta, joka sisältää tietyn palvelun tuottajan palvelut. Vastaa analogisen television kanavaa sillä erotuksella, että yhdessä multipleksissa voidaan lähettää useita kanavia. Tästä syystä multipleksiä kutsutaan myös kanavanipuksi.

MPEG-2; Motion Picture Experts Groupin kehittämä ISO/IEC-standardi 13818, jonka tarkoituksena on tarjota yleinen ratkaisu videon ja audion koodaukseen maailmanlaajuisesti.

NIT Network Information Table; Taulu, joka sisältää tiedot verkon fyysisestä organisoinnista ja verkon ominaisuuksista.

OpenMHP; Digitaalisen television sovelluskehitysympäristö.

PAT Program Association Table; Sisältää tiedon multipleksin jokaisesta ohjelmasta.

PID Program Identifier; Siirtobittivirrassa kulkevien pakettien tunnistet.

PMT Program Map Table; Sisältää tiedon ohjelman komponenteista.

PSI Program Specific Information; MPEG-2-standardissa määritelty palvelutieto.

Sektiot; Tietorakenne, jota käytetään mm. datapalveluiden lähettämiseen siirtobittivirran mukana.

Singleton; Olio-ohjelmoinnin käytäntö, jossa luokasta voidaan luoda vain yksi instanssi tai ilmentymä kerrallaan.

Sovelluskehys; Ohjelmistokehityksessä uudelleenkäyttöä varten suunniteltu systeemi, joka koostuu abstrakteista luokista ja säännöistä, joiden mukaan näistä luodut ilmentymät toimivat.

TCP Transmission Control Protocol; Sääntökokoelma, jota käytetään IP-protokollan kanssa pakettien lähettämiseen Internet-verkossa. TCP huolehtii luotettavasta pakettien käsittelystä siinä missä IP huolehtii niiden kuljetuksesta.

Thin client; Kuluttajan päätelaitteeseen toimitettava sovellus, jonka teknisen koon (kilotavuina) pitäisi olla mahdollisimman pieni joko siirtotien kapeuden, päätelaitteen muistirajoitusten tai muun syyn takia.

Timelineen synkronoidut palvelut; Lisäarvopalvelut, joiden käyttö on mahdollista vain televisio-ohjelmaa katsottaessa siten, että palvelun toiminnallisuus liittyy ohjelman kulkuun ja on ajastettu muuttumaan ohjelman tapahtumien kanssa synkronoidusti.

TLS Transport Layer Security; Paluukanavan turvallisuusjärjestelyissä käytettävä standardi.

UML Unified Modelling Language; Standardi tapa olio-ohjelmoinnin toimijoiden, luokkien ja niiden välisten suhteiden mallintamiseen.

URL Uniform Resource Locator; Internetissä sijaitsevan tiedoston yksilöllinen osoite. URL koostuu käytettävästä protokollasta, palvelimen domain-nimestä ja mahdollisesta tiedoston nimestä palvelimella.

1 JOHDANTO

Suomi aloitti 27.8.2001 ensimmäisenä maailmassa uuteen MHP-standardiin (*Multimedia Home Platform* -ohjelmointirajapintaan) tukeutuvat digi-tv -lähetykset. Elokuussa 2007 Suomessa siirrytään pelkästään digitaalisiin lähetysiin. Digitaalinen siirtotekniikka mahdollistaa entistä laadukkaampien ja monipuolisempien palveluiden tarjonnan. Digitaaliseen ohjelmälähetteeseen voidaan helposti liittää lisämateriaalina esimerkiksi pankki-, kauppa- tai päivityspalveluita. Tampere otti toukokuussa 2005 ensimmäisenä kaupunkina Suomessa käyttöön MHP-palvelut. Portaali tarjoaa tietoa esimerkiksi bussiaikatauluista ja tapahtumista.

MHP-standardin mukaisten palveluiden ja paluukanavan ansiosta käyttäjä pystyy toimimaan vuorovaikutuksessa sovelluksen kanssa. Tämä mahdollistaa useiden palveluiden tuomisen suoraan kuluttajien televisioruutuihin. Kaukosäätimen avulla käyttäjä voi esimerkiksi täyttää lottokupongin tai valita tuotteita ostoskoriinsa.

Tässä työssä tutustuttiin MHP-sovellusstandardiin ja luotiin ohjelmakohtainen sovellus, jolla voidaan paluukanavaa pitkin äänestää lähetettävien ohjelmien laadusta, ohjelmissa esiintyvistä henkilöistä tai asioista. Sovelluskehitysympäristönä käytettiin OpenMHP-alustaa, lisäksi projektin käyttöön oli varattu digiboxi, jossa oli MHP-versio 1.0.2.

Sovellus on kehitetty LähiTV-hankeeseen, jossa kehitetään yhteisöllinen, alueellisia palvelusisältöjä tarjoava digi-tv:n asiointiympäristö. Kokonaisuuteen yhdistetään kaupallisia ja kunnallisia palveluita ja asukkaiden itsensä tuottamia sisältöjä. Palvelun tuottajien sisällöt yhdistetään samaan kokonaisuuteen, joka on käytettävissä eri päätelaitteilla. Tavoitteena on luoda katsojia kiinnostava, paikallisia ohjelmia ja palveluita tarjoava kanava päivittäiseen käyttöön.
[1, 2, 4]

2 DIGITAALINEN TELEVISIO

Digitaalisen television kehittäminen alkoi Euroopassa 1990-luvun alussa. Lähetystekniikan digitalisoinnilla saavutetaan monia etuja. Digitalisointi mahdollistaa lähetyksvirran tehokkaan kaistanleveyden käytön, jolloin yhden analogisen televisiokanavan vaatimaan kaistanleveyteen saadaan mahtumaan 5-6 digitaalista televisiokanavaa. Myös televisiokuvan ja -äänen laatu paranevat. Lähetyksvirran muuttuessa digitaaliseen muotoon, mahdollistuu minkä tahansa datan lähetyks. Suomessa lähetetään rinnakkain edelleen sekä analogisia että digitaalisia lähetyksiä. Digitaalisten lähetyksien vastaanottoon tarvitaan joko erillinen sovitin (ns. set-top-box, digiboxi) tai TV-vastaanotin, jossa digitaalivastaanotin on luontaisena. [3]

2.1 *Historia*

Vuonna 1991 perustettiin digitaalisen television kehittämistä valvova elin, ELG (European Launching Group). Suurimpien eurooppalaisten tv-alan markkinatoimijoiden liittyessä ELG:hen vuonna 1993 käynnistyi Digital Video Broadcasting (DVB)-projekti. Projektin alussa osapuolet muotoilivat yhteiset säännöt, joiden mukaan digitaalista tv-alaa alettiin kehittää. [5, 10]

2.2 *Digitaaliset lisäarvopalvelut*

Tärkein yksisuuntainen lisäarvopalvelu on elektroninen ohjelmaopas (EPG), jonka avulla käyttäjät voivat nähdä säännöllisten kanavien koko tv-ohjelmatarjonnan kerralla.

Broadcasterista riippumattomat asiakaspalvelut ovat palveluja, joita voi käyttää ilman paluukanavaa eivätkä ne siten välttämättä vaadi MHP-digiboxia. Jakelutapoja voivat olla niin boxiin kuin vastaanottimeenkin kiinteästi asennetut sovellukset, lähetyksvirran mukana tulevat ja lähetyksvirrasta ladattavat sovellukset tai maksukortilla tai muulla fyysisellä siirtovälineellä ladattavat sovellukset. On myös mahdollista, että tulevaisuudessa tulee asiakas-palveluja, jotka ladataan internetistä PC:n avulla ja siirretään sitten tv-ohjelman käyttöympäristöön.

Broadcasterista riippumattomat asiakas-palvelin-tyyppiset palvelut, kuten pankkipalvelut, ovat jo pitkälle kehittyneitä ja julkaistuja.

Broadcasterien ohjelmakohtaiset lisäarvopalvelut jakautuvat infopalveluihin ja toiminnallisiin palveluihin. Infopalvelut ovat tyypillisesti lisätietoa ohjelmista, toiminnalliset palvelut taas joko ohjelman toimintalogiikkaa ja dramaturgiaa mukailevia tai irrallisia sovelluksia, jotka on suunniteltu ohjelman ilmeen mukaisiksi. [3]

2.3 *Paluukanava*

Paluukanavalla tarkoitetaan kuluttajan mahdollisuutta kommunikoida lisäarvopalvelun sisällön perusteella Internetin yli. Vaihtoehtoisia keinoja ovat mobiililiittymä (tekstiviesti, GSM-data tai GPRS-data) joko kännykässä tai digiboxissa, sekä puhelin- tai laajakaistaliittymä joko PC:ssä tai digiboxissa. Kaapeliboxeissa vaihtoehtona on lisäksi kaapelimodeemi, joka on monilla kaapeli-tv-kuluttajilla myös tavallinen laajakaistainen Internet-liittymämuoto.

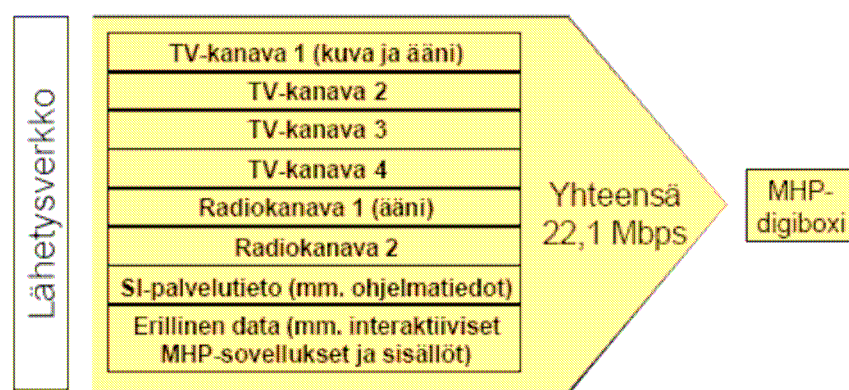
Paluukanavan turvallisuusjärjestelyissä käytetään Internetissä yleisesti käytössä olevaa Transport Layer Security (TLS)-standardia. [5, 6]

2.4 *DVB-standardi*

DVB on standardoitu erikseen maanpäälliseen (DVB-T, Terrestrial), kaapeli- (DVB-C, Cable) ja satelliittijakeluun (DVB-S, Satellite). Lisäksi viimeistelyvaiheessa on liikkuvaan vastaanottoon optimoitu versio (DVB-H, Handheld), jonka kehittämisessä Suomi on ollut erityisen aktiivinen.

DVB pohjautuu MPEG-2-videopakkausstandardiin. DVB:ssä lähetettävän tiedon kannalta kaikki siirtotiet ovat samanlaisia. Lähetettävän tiedon ei tarvitse olla tv-ohjelmaa, vaan se voi olla mitä tahansa bittimuotoista dataa. Yhdessä DVB-kanavanipussa lähetetään normaalisti useita televisiokanavia, radiokanavia, niihin liittyvää palvelutietoa, kuten ohjelmatietoja sekä erillistä dataa (ns. private data). Kuva 1.

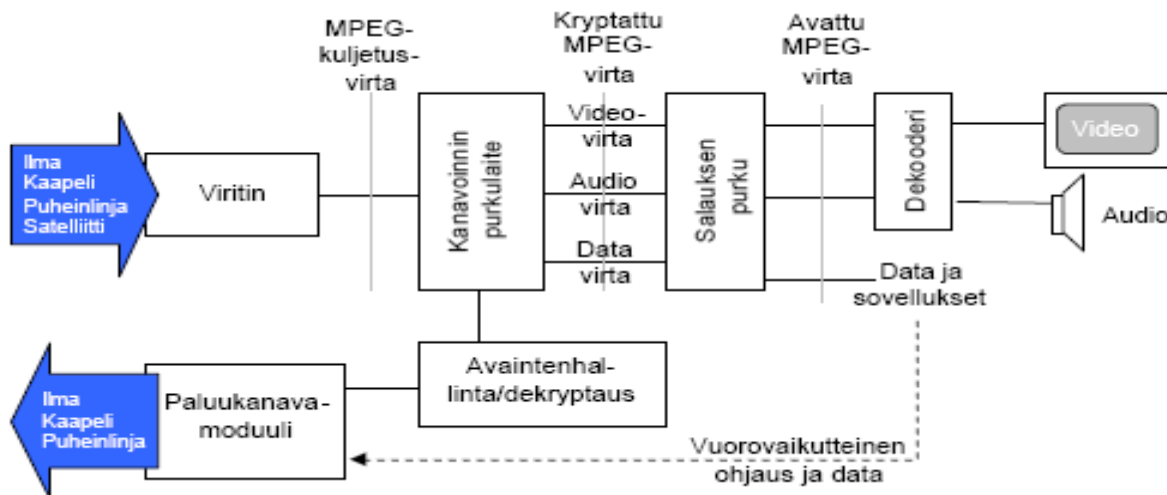
Kaikki vuorovaikutteiset lisäpalvelut toteutetaan erillisen datan piirissä. DVB-standardi on tässä suhteessa joustava – yhden kanavanipun sisällä voidaan tv-kanaville, radiokanaville ja erilliselle datalle määrättyä siirtokapasiteettia muuttaa dynaamisesti. Kanavanipussa voidaan lähettää myös pelkästään erillistä dataa. [6, 7]



Kuva 1 Yhden DVB-kanavanipun sisältö

2.5 Datavirta

Digi-TV-signaali vastaanotetaan virittimeen MPEG-2-kuljetusvirtana kuten kuvassa 16. Kanavoinnin purkulaitteessa virta jaetaan kryptatuiksi video-, audio- ja datavirroiksi. Maksullisten kanavien katsomiseen tarvitaan salauksen purku, ja lopuksi dekooderi muuntaa virran video-, audio- ja datasiinaaliksi. Laitteen toiminnasta ja interaktiivisista sovelluksista lähtee tieto paluukanavan kautta.



Kuva 2. Digitaalisen TV-signaalin vastaanotto

Synkronisuus saadaan vuorovaikutteisiin ohjelmiin esimerkiksi niin, että erityisillä laukaisijoilla ("triggereillä") liitetään ohjelman tiettyyn kohtaan sovelluksia. Teknisesti ja työmäärällisesti helpompaa on kuitenkin tehdä ns. "semisynkronoituja" sovelluksia, joissa sovelluksen ja sen sisällön välittäminen vastaanottimelle ajoitetaan alkamaan samaan aikaan, kun itse TV-ohjelma on lähetyksessä. Lisäksi kaukosäätimen näppäinten toiminta voidaan määritellä uudestaan. Tällöin digisovittimessa oleva kaukosäätimen signaalin vastaanottava ohjelma alkaa toimia uudella tavalla, vaikka kaukosäädin itse jatkaa toimintaansa samalla vanhalla tavalla. [8, 9]

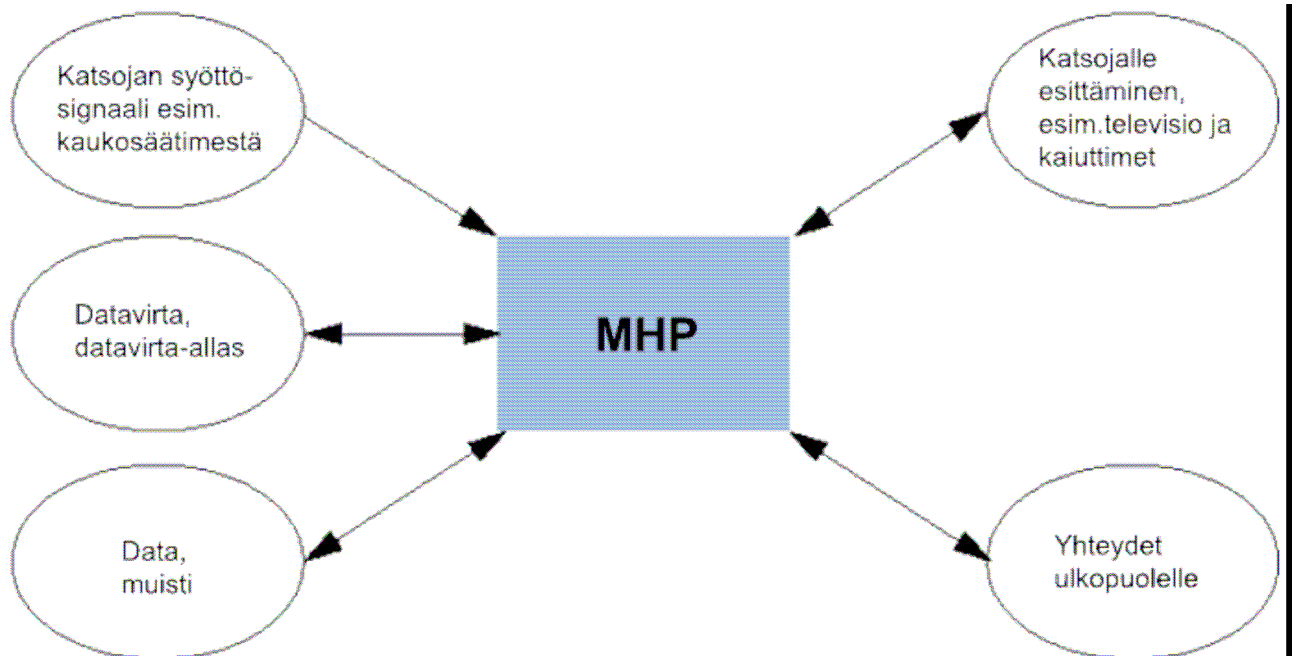
3 MHP-SOVELLUSSTANDARDI

3.1 MHP-standardin historia

Jotta sovellukset toimisivat erilaisissa markkinoilla olevissa laitteissa, on syntynyt tarve kehittää oma standardi vuorovaikuttaville palveluille digi-tv-ympäristössä. DVB-organisaatio alkoi kehittää yhteistä MHP-standardia vuonna 1998. Ensimmäinen MHP-standardi julkaistiin vuonna 2001. MHP on avoin DVB-standardiin sisältyvä ohjelmointialusta, joka perustuu Java-ohjelmointikieleen. MHP-standardi otettiin käyttöön Suomessa 27.8.2001 ensimmäisenä maailmassa.

3.2 Toimintaperiaate

MHP:n toimintatapaa esittää kuvan 3 yksinkertaistettu viitekehys. Siinä digitaalivastaanottoon tuleva data ja datavirta (stream) kulkeutuu MHP-käyttöjärjestelmään. Tulevasta datasta ja datavirroista prosessoidaan esitettäväksi se osa, jonka katsoja syöttösignaaliensa avulla määrittelee. Syöttösignaalit syötetään esimerkiksi kaukosäätimellä tai näppäimistöllä. Dataa ja datavirtoja voidaan myös siirtää laitteiston muistiin tai datavirta-altaisiin. MHP mahdollistaa myös yhteydenoton järjestelmän ulkopuolelle, esimerkiksi puhelin- tai kaapelilinjojen kautta. [5]

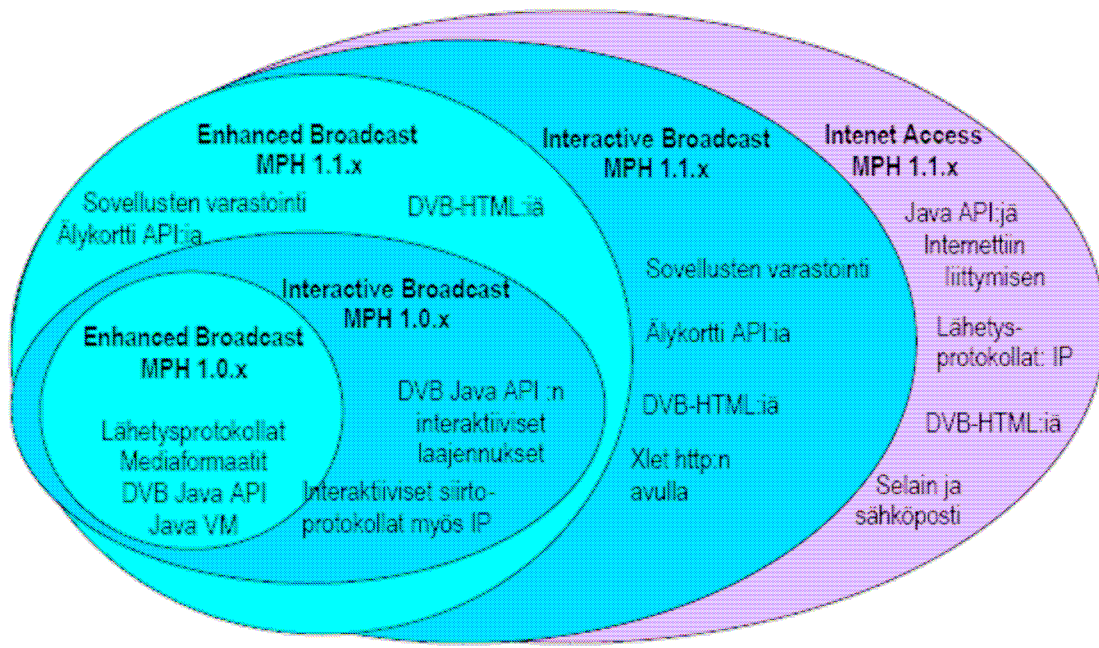


Kuva 3. MHP:n toimintaperiaatteen viitekehys

3.3 MHP -profiilit

MHP:lle on määritelty kolme profiilia, jotka esitellään kuvassa 4. Samasta profiilista julkaistaan uusia versioita. Esimerkiksi Enhanced Broadcast Profilesta on julkaistu versiot 1.0.x ja 1.1.x. Se on myös yksinkertaisin profiili, joka mahdollistaa ilman paluukanavaa toimivien sovellusten ajamisen.

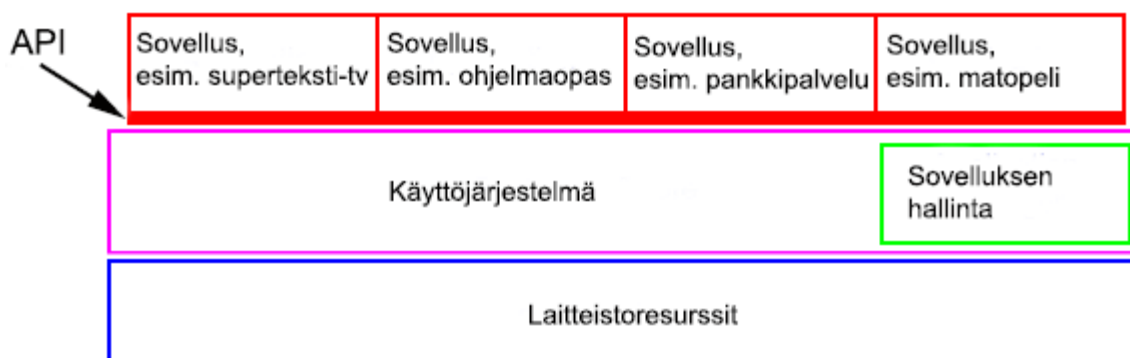
Seuraava profiili on Interactive Broadcast Profile, joka sisältää verkkotoiminnot ja mahdollistaa paluukanavan käytön normaaleja Javan Socket-luokkia hyväksikäyttäen. Laajin on Internet Access Profile, johon kuuluu valmiita sähköposti-, NNTP- ja HTTP-ominaisuuksia. [9, 11]



Kuva 4. DVB-MHP-profiileita ja -versioita

3.4 Arkkitehtuuri

MHP:n arkkitehtuuri rakentuu kolmesta peruselementistä: käytettävissä olevista laitteistoresursseista, käyttöjärjestelmästä ja sovelluksista. Kuva 5 esittää arkkitehtuurimallia.

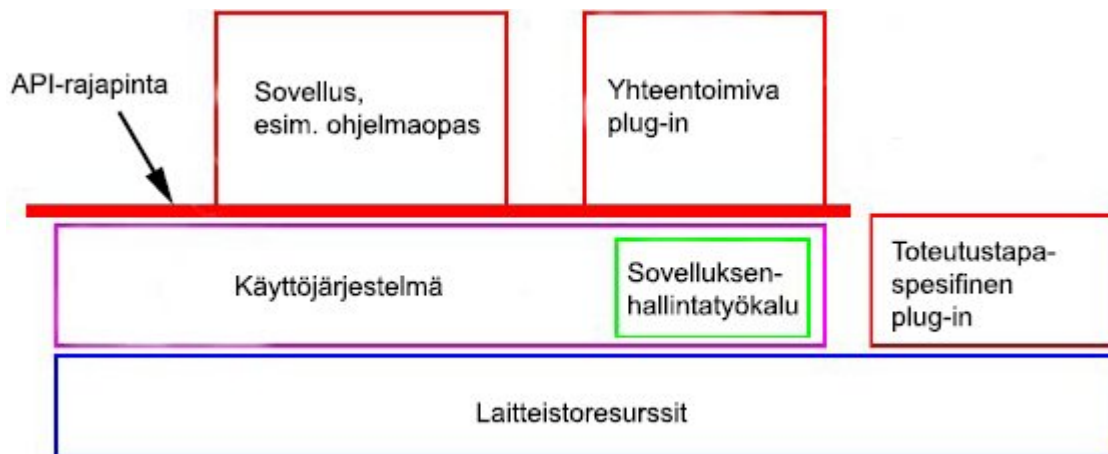


Kuva 5. MHP-arkkitehtuurin peruselementit

Tyypillisiä digitaalivastaanottimessa olevia laitteistoresursseja ovat mm. keskusprosessori, muisti ja grafiikkasuoritin. Käyttöjärjestelmän tehtävänä on hallita ja palvella digitaalivastaanottimessa ajettavia ohjelmia. Käyttöjärjestelmä sisältää lisäksi sovelluksen hallintatyökalun (application manager), joka säätelee suoritettavien sovellusohjelmien elinkaarta.

Käyttöjärjestelmän ja sovelluksien välissä on MHP-standardissa määritelty API (Application Program Interface)-ohjelmistorajapinta. Se yhdessä käyttöjärjestelmän kanssa muodostaa tuntumattoman rajapinnan digitaalivastaanottimen laitteistoresursseihin. Tämä tarkoittaa sitä, että sovellukset eivät hyödynnä laitteistoresursseja itsenäisesti, vaan käyttöjärjestelmän välityksellä. Standardoitu API-rajapinta mahdollistaa myös sen, että sovellukset saavat käyttöönsä yhtenäisen ja standardisoidun rajapinnan riippumatta siitä, mikä käyttöjärjestelmä digitaalivastaanottimessa on tai minkälaisiin teknisiin ratkaisuihin se perustuu.

Jotta sovellusohjelmat toimisivat kaikissa erilaisissa vastaanottimissa, MHP-standardi määrittelee toimintatavat lisätoiminnallisuuden eli ns. plug-in asentamiseen. Plug-in asentamisesta vastaanottimeen päättää aina palvelun käyttäjä. Lisätoiminnallisuudet voidaan toteuttaa kahdella eri tavalla, kuten kuva 6 esittää. Toteutustavaltaan spesifinen plug-in tehdään osaksi käyttöjärjestelmää, jolloin sen toimivuus rajoittuu tietyn tyyppisiin vastaanottimiin. Vastaavasti yhteensopivat plug-in:t suoritetaan MHP-sovellus-ohjelmien kaltaisesti, jolloin ne toimivat kaiken tyyppisissä MHP ympäristöissä. [5, 12]



Kuva 6. MHP:ssä lisätoiminnallisuudet voidaan tehdä yhteensopiviksi tai toteutustavaltaan spesifisiksi

3.5 Syötteet

MHP-standardissa on määritelty käyttäjän syötteille kolme laitetta: kaukosäädin, näppäimistö ja hiiri. Laittevalmistajille pakollisia syötelaitteita on vain kaukosäädin. MHP-standardin kaikki sallitut painikkeet on määritelty luokassa `org.havi.ui.event.HRcEvent`.

MHP määrittelee laitteiden kaukosäätimien minimivaatimukset. Kaukosäätimistä tulisi löytyä vähintäänkin seuraavat näppäimet:

- nuolinäppäimet ylös, alas, vasemmalle ja oikealle
- OK
- TXT
- numeronäppäimet 0-9 sekä
- värinäppäimet järjestyksessä punainen, vihreä, keltainen, sininen.

OK- ja TXT-näppäimet voidaan kaukosäätimessä kuvata jollain toisellakin tavalla, esimerkiksi symbolilla. [8, 13]

3.6 Katsojan tunnistaminen

MHP-standardi antaa katsojalle mahdollisuuden syöttää tietojaan vastaanottimen muistiin, jolloin sovellukset voivat niitä käyttää. Tällaisia tietoja ovat käytettävä kieli, sähköpostiosoitteet, nimitiedot yms. Tietojen syöttäminen on vapaaehtoista, ja sen tarkoituksena on parantaa käytettävyyttä esimerkiksi lomaketietojen esitähän yhteydessä. Käyttäjän tiedot voidaan myös hakea ns. älykortilla, joka on liitetty digitaalisovittimeen. MHP-sovellus voi lukea älykortilta nämä tiedot ja varmistaa henkilöllisyyden käytettäessä esimerkiksi pankkisovelluksia. Lisäksi käyttäjä joutuu tällöin syöttämään myös henkilökohtaisen PIN-koodin, joka toimii eräänlaisena allekirjoituksena. Tiedot välitetään paluukanavan kautta palveluntarjoajalle, jolloin niiden oikeellisuus ja eheys tarkistetaan. [14]

3.7 Muistivaatimus

Vastaanottimien vähimmäismuistikapasiteetti on määritelty, jotta sovellusten toimivuus kaikissa vastaanottimissa kyetään takaamaan. NorDig II -standardin asettamat muistivaatimukset ovat

- 16 Mt RAM-muistia
- 4 Mt video-RAM-muistia
- 8 Mt Flash-muistia

Vastaanottiin asennettu kiintolevy ei ole pakollinen, mutta suositeltava. [5]

3.8 Turvallisuus

MHP-määrittelyssä on tuki sovellusten turvallisuusmallille. Esimerkiksi paluukanavaa käyttävät sovellukset täytyy allekirjoittaa sähköisesti, jotta vastaanotin ja katsoja voivat varmistua sen alkuperästä. Muuten ohjelmavirtaan jossain kohdassa liitetty sovellus voisi esimerkiksi soittaa katsojan tietämättä maksulliseen puhelinnumeroon. Samoin sovelluksen pitää olla allekirjoitettu, jotta se voisi hyödyntää vastaanottimen pysyväis-muistia tai vaihtaa kanavaa. Allekirjoitetuille sovelluksille määritellään yksityiskohtaisesti oikeudet toimenpiteille, joita ne voivat suorittaa. [15]

4 MHP-SOVELLUS

MHP on avoimella Java-kielellä kirjoitettavien digitaalisten televisio-ohjelmien käyttöalusta. Ohjelmoinnin kannalta MHP:n ongelmat liittyvät vaadittavan koodin keveyteen verrattuna esimerkiksi PC:lle kirjoitettavaan koodiin. Latausaikojen on oltava mahdollisimman lyhyitä ja sovelluksen on mahdollista datakaruselliin lähetysvirran joukkoon, jossa samasta tilasta kilpailee monta muutakin datasisältöä. Koska kyseessä on ns. thin client, ohjelmoinnilta vaaditaan lisää laatutasoa (tiukkuutta, vähämuistisuutta). [3, 16]

4.1 MHP-sovelluksen elinkaari

MHP-sovellus voi käynnistyä joko automaattisesti tai käyttäjän pyynnöstä. Aktiivisena ollessaan se voi piirtää tv-ruudulle grafiikkaa ja olla vuorovaikutuksessa katsojan kanssa. Sovellus voidaan myös asettaa ns. paused-tilaan, jolloin se pysyy käynnissä taustalla. Tällöin sovellus on nopeasti saatavissa jälleen aktiiviseksi. Useimmiten sovelluksen elinkaari loppuu kanavanvaihdon yhteydessä. On myös mahdollista signaloida samaa sovellusta usealla kanavalla siten, että sovellus säilyy ajossa kanavanvaihdon jälkeen. Sovelluksen lopetuksen jälkeen vastaanottimen muistiin jäävät vain sovelluksen mahdollisesti pysyväismuistiin kirjoittamat tiedot. Itse sovelluksia voidaan tallentaa pysyväismuistiin vasta MHP-määrittelyn version 1.1 mukaisissa seuraavan sukupolven vastaanottimissa. Jotkut MHP-vastaanottimet tarjoavat katsojalle erikseen valikon, josta sovelluksia voidaan hallita. Katsoja näkee missä tilassa eri sovellukset ovat ja voi käynnistää, piilottaa (asettaa paused-tilaan) ja lopettaa niitä. MHP-vastaanotin voi asettaa sovelluksen paused-tilaan halutessaan näyttää omaa ruutugrafiikkaansa. Paused-tilassa sovelluksen tulisi vapauttaa mahdollisimman paljon käyttämistään resursseista. Tärkeintä kuitenkin on, että sovellus piilottaa itsensä näkyvistä. Myös palveluvalikot voivat käyttää hyväkseen käynnistettävien sovellusten paused-tilaa. Tärkeimmät sovellukset voidaan pitää koko ajan taustalla paused-tilassa, josta ne käynnistyvät nopeammin ja ovat näin nopeammin katsojan käytettävissä. [15]

4.2 Sisältöformaattit

MHP-vastaanottimet kykenevät esittämään yleisimpiä bittikarttaisia kuvaformaatteja, kuitenkin standardissa määritellyin rajoituksin. Kuvien pikseliskaalaus sekä väriavaruuden ja gammakertoimen muutokset eivät ole sallittuja. Tuetut bittikarttaiset kuvaformaattit ovat:

- JPEG
- PNG
- GIF
- MPEG-2 I-Framet

Video- ja audioklippien esittämistä varten tuettuja formaatteja ovat:

- MPEG-2 Video ”drips”
- MPEG-1 Audio (Layer 1 & 2)

Tekstin koodaukseen käytetään UTF-8 -standardia, joka on modifioitu Java yhteensopivaksi. Lähetysvirrassa tulevan videokuvan ja äänen formaatit on määritelty MHP-standardissa tietyin rajoituksin ja lisäyksin. Tuetut formaatit ovat:

- MPEG-2 Video (25Hz)
- MPEG-1 Audio (Layer 1 & 2)

Ohjelmatekstitystä varten on määritelty kaksi vaihtoehtoista mekanismia, jotka ovat:

- DVB subtitle
- Teletext

[12]

4.3 MPEG-2-standardi

MPEG-2 -siirtobittivirta on varsinainen tietovirta, jota lähetetään multiplekseissa kaikelle kansalle vastaanotettavaksi. Se koostuu jatkuvasta tietopakettivirrasta, jossa audio-, video- ja datalähetykset kulkevat limittäin.

MPEG-standardi perustuu neljään elementtiin: system, video, audio ja testaus. System määrittelee, kuinka audio- ja videosaatimet synkronoidaan eli tahdistetaan ja miten multipleksaus tapahtuu. Videoelementti määrittelee menetelmän, jolla kuvaa pakataan eli kompressoitetaan, ja vastaavasti äänielementti määrittelee äänen pakkaustavan. Testaelementti määrittelee kodekin bittivirran ominaisuudet ja dekodeusprosessin.

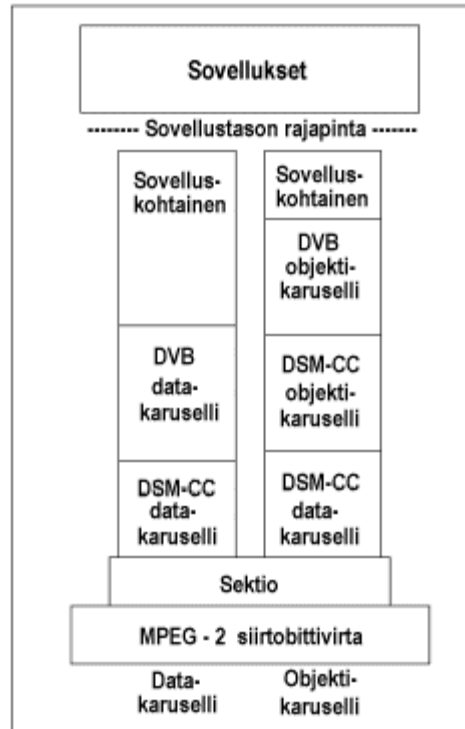
Analogisen video- ja äänisaatimen muuttaminen biteiksi tuottaa suunnattoman määrän dataa, jonka lähettäminen ilman pakkausta olisi taajuuskaistojen tuhlausta. MPEG-2-kompressiomenetelmässä kuva- ja ääni-informaatiosta poistetaan toistoa. Kuvan yksityiskohtien värimäärää vähennetään ja liikkuvan kuvan terävyyttä vähennetään. Kuvan välityksessä lähetetään vain kuvassa tapahtuvat muutokset, jolloin vältetään koko kuvan lähettäminen uudelleen. Äänen kompressoinnissa alkuperäisestä materiaalista poistetaan mm. taajuusalueita, joita ihmiskorva ei pysty kuulemaan. Pakkauksessa menetetään siis osa alkuperäisestä informaatiosta, mutta tämä kadotettu informaatio on sellaista, jota ihmissilmä tai -korva ei pysty huomaamaan.

Siirtobittivirrassa kulkevat paketit erotellaan toisistaan paketin otsikossa kulkevan PID (program identifier)-tunnisteen avulla. Jotta vastaanotin tietää millä PID-arvolla ohjelmaan kuuluvat paketit siirtobittivirrassa kulkevat, siihen on lisätty SI (service information)-ohjaustietoja.

DVB-bittivirrasta löytyy yleensä yhdeksän taulua, joista seuraavat neljä ovat pakollisia: PAT (program association table), joka sisältää tiedon multipleksin jokaisesta ohjelmasta, PMT (program map table), joka sisältää tiedon mitä komponentteja mikäkin ohjelma sisältää ja millä PID-tunnuksella komponentit lähetetään, CAT (conditional access table), joka sisältää tiedot salauksen purkuun, jos salausta käytetään sekä NIT (network information table), joka sisältää tiedot verkon fyysisestä organisoinnista ja verkon ominaisuuksista. Nämä neljä pakollista taulua muodostavat ohjelman palvelutiedot, PSI (program specific information). [12, 17, 19]

4.4 Data- ja objektikaruselli

Karusellit ovat digitaalisiin datalähetyksiin käytettäviä protokollia, joiden avulla informaatiota lähetetään toistuvasti tekstitelevisioon tapaan palvelimelta useille vastaanottajille samanaikaisesti. Lähetettävä informaatio voi olla minkä tyyppistä tahansa, vaikkapa television ohjelmatietoja tai pelisovelluksia.



Kuva 7. Karusellien sisäinen rakenne

Datakarusellin protokollapino on yksinkertaisempi kuin objektikarusellin. Kuvan 7. DSM-CC datakaruselliosuuden lisäksi siinä on ainoastaan DVB-datakaruselliosuus, jonka päälle voidaan toteuttaa sovelluksittaisia osia.

Objektikaruselli on datakarusellia monimutkaisempi kokonaisuus. Sektioiden päällä on DSM-CC-datakaruselli-osuus samoin kuin datakarusellissa. Objektikaruselli käyttää datakarusellin data- ja kontrolliviestejä varsinaisen informaation kuljettamiseen. DSM-CC -objektikaruselliosuus lisää datakaruselliviestien päälle hierarkkisen nimiavaruuden, jonka avulla voidaan hyödyntää objektikarusellin tarjoamaa virtuaalista tiedostojärjestelmää. DVB on lisännyt tännekin televisiojärjestelmiä varten omia toiminnallisuksiaan samoin kuin datakarusellissa.

Datakarusellin avulla voidaan lähettää ainoastaan tiedostoja yhden hakemiston sisällä, mutta sillä ei voida tarjota objektikarusellin kaltaista hierarkkista hakemistorakennetta. Objektikaruselli antaa mahdollisuuden luoda virtuaalisen tiedostojärjestelmän, jonka puitteissa tietoihin viitataan loogisilla nimillä. Objektikaruselli mahdollistaa myös viittaukset toisiin palveluihin. Datakarusellin avulla voidaan järjestää tietoja ns. ryhmiin ja superryhmiin. Tämä helpottaa tietojen kokoamista loogisiin kokonaisuuksiin.

Objektikarusellin mukana lähetettävä tieto voidaan pakata tehokkaammin kuin datakarusellissa, koska objektikarusellin tiedostoja voidaan sijoittaa useita yhteen datakarusellin dataviestiin. Datakaruselli on protokollana huomattavasti objektikaruselliä kevyempi. JavaTV API on yksi kokonaisuus, joka sisältää myös rajapinnat karuselli-informaation hyödyntämiseen. Se onkin otettu osaksi DVB MHP -määrittystä. [17]

4.5 Tiedostojen käsittely

Tiedostojen latausaika ei digi-tv-ympäristössä määrity suoraan yksittäisen tiedoston tiedostokoon ja käytettävissä olevan kaistanleveyden suhteen. Usein suurin osa odotusajasta kuluu vastaanottimen odottaessa oikean tiedoston saapumista kohdalle. Odotusaikaan vaikuttaa siis kaistanleveyden lisäksi koko objektikarusellin yhteenlaskettu tiedostokoko. Esimerkiksi 256 kB karusellin pyörähdysaika 128 kb/s:n kaistanleveydellä on noin 16 sekuntia. Objektikaruselliin voidaan määritellä erityisasetuksia, esimerkiksi sovelluksen aloitukseen tarvittavat tiedostot voidaan asettaa lähetettäväksi kahden sekunnin välein. Jokin kriittinen tiedosto voidaan asettaa välittömästi päivittyväksi eli seuraavaksi karuselliin lähteväksi tiedostoksi. Jokin harvemmin tarvittava tiedosto saatetaan lähettää vain joka toisella karusellin pyörähdyksellä. Objektikaruselli saattaa sisältää myös vihjeitä siitä, kuinka vastaanottimen tulisi säilyttää tiedostoja välimuistissaan. Muuttumattomat tiedostot voidaan merkitä staattisiksi, jolloin vastaanotin voi ladata ne kerralla välimuistiinsa, ja ne ovat näin aina nopeasti sovelluksen saatavilla. [15]

4.6 Syötteiden käsittely

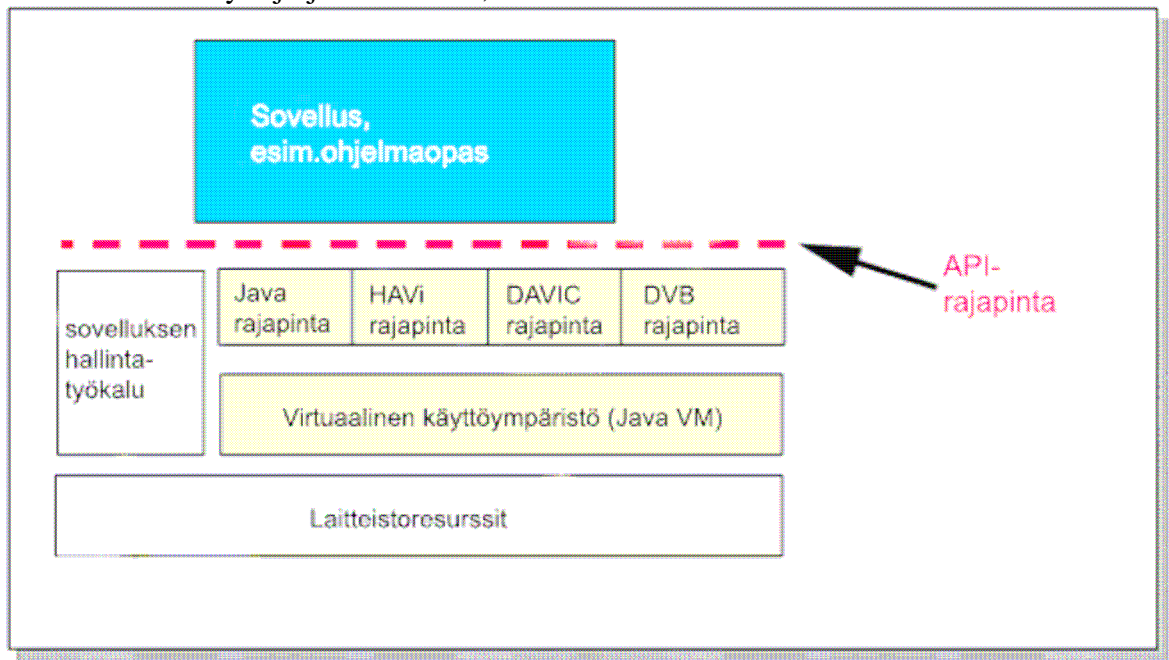
MHP-standardi mahdollistaa käyttäjän syötteiden käsittelyn neljällä eri tavalla. Nämä ovat seuraavat:

1. Standardien Java AWT-tapahtuman kuuntelijoiden avulla
2. Kuten edellisessä, mutta sallien tietyn sovelluksen yksinoikeuden joihinkin tapahtumiin.
3. org.dvb.event-API:n avulla.
4. Kuten edellisessä, mutta sallien tietyn sovelluksen yksinoikeuden joihinkin tapahtumiin.

Java AWT:n tapahtuman kuuntelumallille on tarvittu vaihtoehtoinen malli, koska AWT tarvitsee koko ajan jonkin käynnissä olevan komponentin, jolla on sovellusympäristön fokus ja joka kuuntelee tapahtumia. Tämä malli vie digi-tv-päätelaitteessa monessa tapauksessa liikaa kapasiteettia muulta prosessoinnilta ja siksi vaihtoehdoksi on kehitetty oma org.dvb.event-paketti ja sen UserEventListener-luokka. Tämän mallin mukaisesti vastaanotetut tapahtumat eivät siis ole AWT-tapahtumia, vaan luokan org.dvb.event.UserEvent-ilmentymiä. Jotta MHP-sovellus kykenee käsittelemään käyttäjän syötteitä, tulee siinä olla luotuna syötteiden varasto eli org.dvb.event.UserEventRepository-olio, johon määritellään kaikki ne syöteryhmät, joita kyseisen sovelluksen tulee olla valmis vastaanottamaan. Pääsy syötteisiin pyydetään erilliseltä org.dvb.event.EventManager-oliolta, joka on singleton-tyyppinen olio ja siten yhteinen kaikille päätelaitteissa toimiville sovelluksille. [12, 13]

4.7 Sovellustyypit

Tässä työssä keskitytään DVB-J- (Digital Video Broadcasting - Java) sovellustyyppiin. DVB-J on yleisin digitaalitelevision ympäristöön spesifioitu Java-alusta. Sen toiminnallisuus on määritelty MHP-standardissa. DVB-J:n ytimenä on virtuaalinen käyttöympäristö, Java VM (Java Virtual Machine), jolla eri sovellusohjelmien suorittaminen on mahdollista ilman, että näitä ohjelmia tarvitsisi erikseen kääntää tai liittää osaksi vastaanottimen käyttöjärjestelmää. DVB-J on osa MHP-arkkitehtuurin käyttöjärjestelmätasoa, kuten kuvasta 8 ilmenee.



Kuva 8. Keltaisella kuvattu DVB-J koostuu Java VM:stä sekä eri rajapinnoista /74/

DVB-J koostuu myös joukosta ohjelmointirajapintoja. Nämä rajapinnat ja niiden toiminnallisuudet voidaan luokitella seuraavasti:

Java

- Javan perusraajapinnat (lang, util, beans ...)
- esitysraajapinnat (AWT, JMF)
- palvelunvalintaraajapinnat (JavaTV)

HAVi (Home Audio Video Interoperability)

- esitys- ja käyttöliittymäraajapinnat

DAVIC (Digital Audio Visual Council)

- maksu-tv-raajapinnat
- infrastruktuurirajapinnat
- viritysraajapinnat

DVB (Digital Video Broadcasting)

- laajennukset ja rajoitukset Java-raajapintoihin
- tiedonsaantirajapinnat
- palveluntieto- ja -valintaraajapinnat
- I/O-laitteiden rajapinnat
- turvallisuusraajapinnat
- muut rajapinnat

Toinen digitaalisen television sovellusohjelmatyyppi on DVB-HTML (Digital Video Broadcasting – HyperText Markup Language). Se on digitaaliseen televisioympäristöön kehitetty ohjelmointikieli, joka koostuu HTML-kieleen liittyvistä teknologioista. [10, 11, 18]

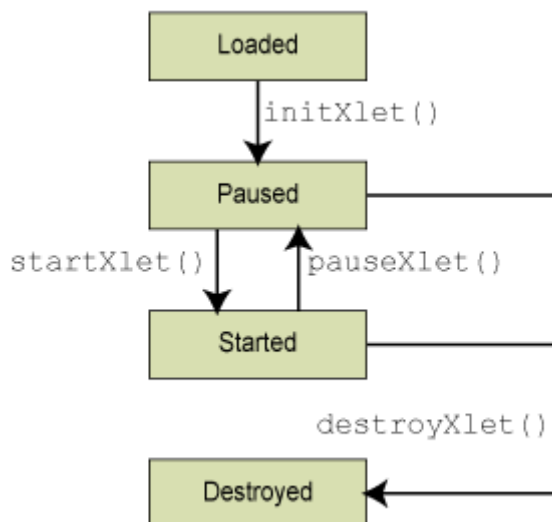
4.8 *JavaTV API:n kategoriat*

JavaTV API on jaettu seuraaviin API-pääryhmiin. [20]

- Service and Selection APIs
 - Tarjoaa mahdollisuuden useampien palveluiden yhtäaikaiseen tarkisteluun.
- Broadcast Pipeline APIs and JMF
 - Mahdollistaa siirtoteistä riippumattomien aikaperusteisten medioiden näytön, sekä kontrolloi ja mahdollistaa median käsittelyn.
- Broadcast Data API
 - Datan tiedonsiirtoon. Kerrottu tarkemmin kappaleessa 4.4, data- ja objektikaruselli.
- Application Lifecycle APIs (Xlet)
 - Digitaalisessa televisiossa ajettavia sovelluksia kutsutaan Xleteiksi. Kappaleessa 4.9 kerrotaan Xleteistä tarkemmin.

4.9 *Xlet-rajapinta*

MHP-sovellukset eivät ole itsenäisiä sovelluksia vaan tietoturvasyistä päätelaite kontrolloi niiden toimintaa erityisen Xlet-rajapinnan avulla. MHP-päätelaitteessa toimiva hallintaohjelmisto on vastuussa kaikkien sovellusten käynnistyksestä, pysäyttamisestä ja resurssien jakamisesta eri sovellusten välillä. Mikäli lähetysvirrasta on ladattavissa MHP-sovellus, on tästä kerrottu lähetysvirran sovellusinformaatiotaulussa ns. AIT-sovellustietotaulussa. Taulussa on kerrottu hallintaohjelmistolle sovelluksen Xlet-luokka. Kyseessä on luokka, joka toteuttaa Javax.tv.xlet.Xlet-rajapinnan ja siten metodit sovelluksen käynnistämiseen ja pysäyttämiseen. Näiden metodien kautta hallintaohjelmisto kontrolloi sovellusta. Xlet-rajapinta mahdollistaa myös sovelluksen jatkamisen pysäyttämisen jälkeen samasta tilasta. Tästä ominaisuudesta on hyötyä televisioympäristössä, jossa katsoja voi haluta välillä palata katsomaan televisiokuvaa kokonaisuudessaan ja palata tämän jälkeen uudelleen sovellukseen. MHP-sovellus voi olla neljässä tilassa: 1. Ladattu lähetysvirrasta 2. Pysäytetty 3. Käynnistetty 4. Tuhottu / poistettu.



Kuva 9 Xletin tilakonemalli.

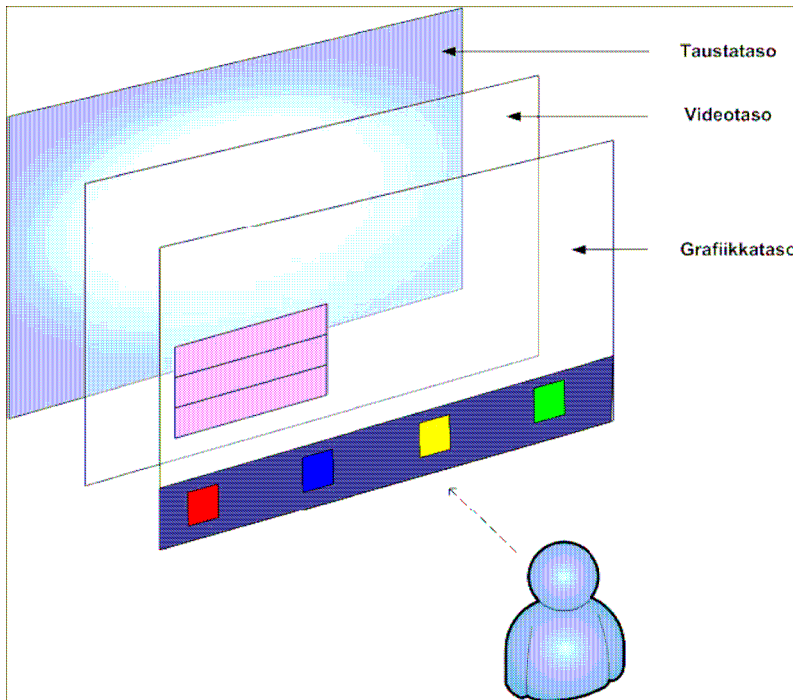
Kuvassa 9 on esitetty Xlet-sovelluksen tilakoneen malli. Xlet voi vaihtaa tilaa joko sovellusmanagerin toimesta tai itsenäisesti. Mikäli sovellus vaihtaa tilaa itsenäisesti, sen on signaloitava sovellusmanageria tästä. Xlet on myös voitava keskeyttää ja poistaa käytöstä milloin tahansa. Taulukossa 1 on selvitys Xletin tilasiirtymistä ja eri tilojen merkityksistä. [18, 19, 20]

Tila	Kuvaus
Loaded	<p>Xlet on ladattu mutta ei alustettu. Tilaan tullaan seuraavassa tapauksessa:</p> <ul style="list-style-type: none"> • Uusi Xlet-instanssi on luotu <code>new</code>-operaattorilla. Tyypillisesti tässä vaiheessa Xlet tekee pieniä alustustoimenpiteitä. Poikkeuksen tapahtuessa Xlet menee Destroyed-tilaan.
Paused	<p>Xlet on alustettu ja valmis käynnistymään. Xletin ei pitäisi varata tai käyttää jaettuja resursseja tässä tilassa. Tilaan tullaan seuraavissa tapauksissa:</p> <ul style="list-style-type: none"> • Loaded-tilasta sen jälkeen, kun suoritus on palannut onnistuneesti suoritetusta <code>Xlet.initXlet()</code>-metodista. • Active-tilasta sen jälkeen, kun suoritus on palannut onnistuneesti suoritetusta <code>Xlet.pauseXlet()</code>-metodista. • Active-tilasta ennen suorituksen palaamista Xletiin onnistuneesti suoritettun <code>XletManager.paused()</code>-metodin jälkeen.
Started	<p>Xlet on suorituksessa normaalisti ja sen tarjoama palvelu on käytössä. Tilaan tullaan seuraavassa tapauksessa:</p> <ul style="list-style-type: none"> • Paused-tilasta sen jälkeen, kun suoritus on palannut onnistuneesti suoritetusta <code>Xlet.startXlet()</code>-metodista.
Destroyed	<p>Xlet on vapauttanut varaamansa resurssit ja lopettanut toimintansa. Tilaan tullaan seuraavissa tapauksissa:</p> <ul style="list-style-type: none"> • Suoritus on palannut onnistuneesti suoritetusta <code>Xlet.destroyXlet()</code>-metodista. • Suoritus on palannut Xletiin onnistuneesti suoritetusta <code>XletManager.destroyed()</code>-metodista.

Taulukko 1 Xlet-tilakoneen tilat ja siirtymät.

4.10 Grafiikka

MHP-laitteissa on kolme eri grafiikkatasoa. Kuva 10. Taustatasolle voidaan asettaa koko ruudun kokoinen täysvärinen I-Frame-kuva tai taustaväri. Sen päällä sijaitsevalla videotasolla voi olla koko ruudun kokoinen tai skaalattu videokuva. Päälimmäisenä on grafiikkataso, jolle kaikki muu sovelluksen piirtämä grafiikka sijoitetaan.



Kuva 10. Digi-tv:n näytön rakenne MHP-sovelluksissa

Grafiikkatasoja voi olla useampia kuin yksi mikäli halutaan esimerkiksi käyttää tekstitykselle omaa grafiikkatasoa ja valikoille omaansa. Grafiikkatasojen määrä riippuu päätelaitteesta ja suoritettavasta MHP-ohjelmasta. Taustataso voi näyttää tiettyä väriä tai yhtä pysäytettyä MPEG-I-kuva, joka voi toimia tarvittaessa taustakuvana. Grafiikkatasoja ohjaa MHP-mallissa `org.havi.ui.HScreen`-luokka, josta päätelaite luo käynnistettäessä ilmentymän jokaiselle siihen kytketylle näytölle. Tyypillisesti päätelaitteeseen on kytkettynä yksi televisio, joten `HScreen`-ilmentymä luodaan tätä televisiota varten.

Sovelluksella voi olla vain yksi `HScene`-ilmentymä, joka vastaa `Java.awt.Frame`-luokkaa. Ilmentymään lisätään `org.havi.ui.HComponent`- ja `org.havi.ui.HContainer`-olioita, jotka muodostavat varsinaisen käyttöliittymän. Ilmentymä pyydetään erilliseltä `org.havi.ui.HSceneFactory`-luokalta, jonka tehtävä on tarjota päätelaitteen ja television ominaisuuksiin mahdollisimman sopiva `HScene`. `HScene`-ilmentymän kautta sovelluksella on mahdollisuus muuttaa televisioon liittyviä graafisia ominaisuuksia kuten kuvasuhdetta ja grafiikkatason läpinäkyvyyttä.

Piirtotasot voidaan tehdä läpinäkyviksi alfa-kanavan avulla, jolloin jokaiselle pikselille lisätään yksi ylimääräinen arvo ilmaisemaan pikselin läpikuultavuusasteen. MHP-standardissa vaaditaan kolme läpikuultavuustasoa: 0% (ei läpikuultavuutta), 30% ja 100% (täysin läpinäkyvä). Vastaanottimet saattavat tukea useampiakin läpikuultavuusarvoja.

Kaikkien grafiikkatasojen resoluutio on 720x576 pikseliä. Kuva-alaa kuitenkin venytetään valitusta kuvasuhteesta riippuen vaakasuunnassa, jolloin lopulliset pikselit eivät ole neliön muotoisia. Vastaanottimet näyttävät kuvien ja grafiikan värit niin hyvin kuin osaavat. MHP-standardissa on määritelty 188 värin vakiopalettei, joka kaikkien vastaanottimien on kyettävä esittämään.

MHP tukee myös MPEG-2 P-Frame-kuvien syöttämistä videodekooderille, mikä mahdollistaa hitaan animaation (korkeintaan kaksi kuvaa sekunnissa) videokuvista. Myös HAVI-käyttöliittymäkomponenteilla on mahdollista rakentaa alkeellinen animaatio.

MHP-määrittelyn Java-versioon ei kuulu Java-sovelluksissa yleisesti käytettyjä JFC- ja Swing-luokkia, vaan osana DVB-MHP-standardia ovat HAVI UI -komponentit. HAVI määrittelee valmiin arkkitehtuurin käyttöliittymäkomponenteille. HAVI-komponentteja käyttävä sovellus toimii tyypillisesti hitaammin kuin sovellus, jonka käyttöliittymäkomponentit on toteutettu itse Java-kielellä, kuten äänestys-sovelluksessa.

Graafisen käyttöliittymän toteutus digitaalisessa televisiossa pohjautuu Java AWT (Abstract Windowing Toolkit) -työkalupakkiin. MHP-standardissa on määritelty tuettavat Java.awt -luokat. Java AWT:n toiminnallisuuksia on lisätty DVB-organisaation määrittelemien org.dvb.ui ja org.dvb.event -luokkien avulla. Laajennukset on räätälöity televisioympäristöön sopiviksi.
[5, 8, 10, 13, 18, 20]

4.11 Ääni

Sovellukset voivat soittaa ääniä tiedostoista. Äänitiedostojen formaatti on MP2 (MPEG-2 Audio Layer 2 compression). Sovellus ei voi säätää tv-lähetykseen kuuluvan äänen voimakkuutta, mutta voi kyllä tarvittaessa mykistää sen. Suuret äänitiedostot hidastavat sovelluksen latautumista karusellista ja kuluttavat enemmän vastaanottimen muistia. [15]

5 Äänestyssovellus

Tutkintotyön ohessa on osallistuttu MHP-standardin mukaisen äänestyssovelluksen luomiseen. Sovellus lähetetään käyttäjälle ohjelmälähetteessä, josta se voidaan ladata digiboxiin. Sovellus luo äänestysvalikon ylläpitäjän lähettämien tietojen perusteella. Annetuista vaihtoehtoista katsoja voi äänestää haluamaansa vaihtoehtoa.

5.1 Sovellusympäristö

Äänestys-sovellus on luotu tekstieditoreilla - mm. Textpad 4, joka sisältää Java-kääntäjän – ja testattu OpenMHP 1.0 sovelluskehitysympäristössä. Kehitysympäristön toimintaan saattaminen edellytti mm. J2SE SDK:n, JavaTV:n, sekä OpenMHP:n asennusta. Sovelluksen kehityksessä on käytetty vain maksuttomia ohjelmia, joita voi ladata internetistä. Osa ohjelmista edellyttää rekisteröitymistä.

Seuraavassa ohje ympäristön asennukseen. tarkemmat ohjeet ja päivitykset tietoihin voi ladata ArviDin ja OpenMHP:n kotisivuilta (lähteet 15 ja 21).

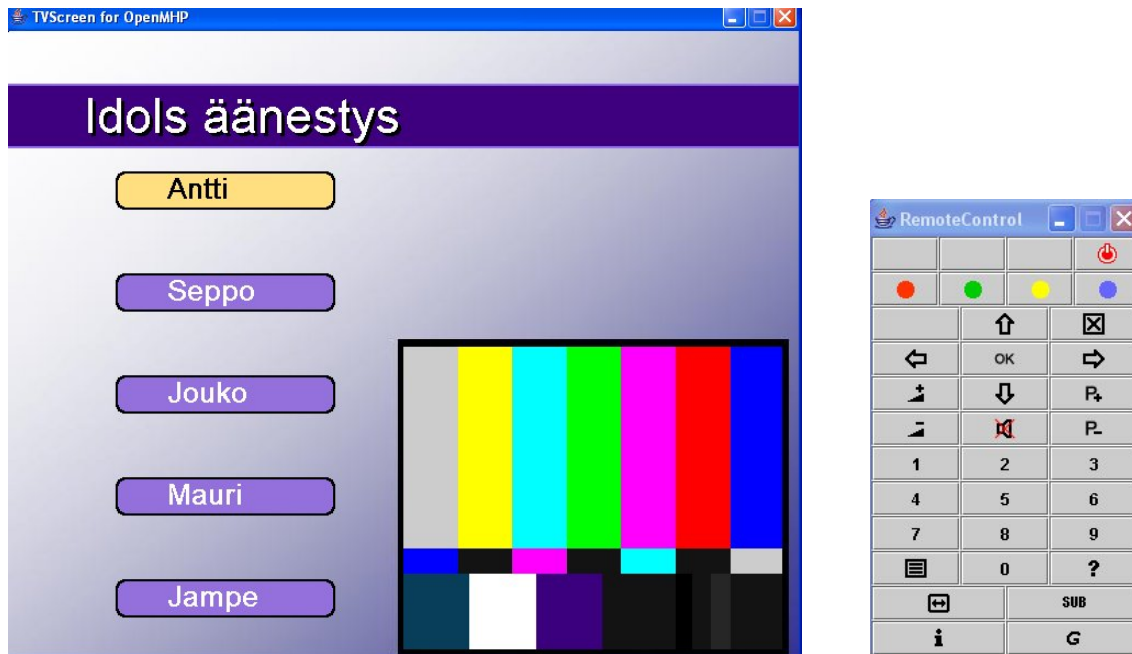
- Asenna J2SE SDK.
- Asenna Java 1.4. JRE
- Asenna OpenMHP
- Asenna Java-TV paketti samaan kansioon kuin OpenMHP.
- Tarkista, että j2sdk\bin kansio on ympäristömuuttujissa (Windows XP:ssä lisääminen tapahtuu valitsemalla Start Menu -> Control Panel -> System -> Advanced -> Environment Variables. Valitse edit ja lisää PATH muuttuja ja sen arvoksi C:\j2sdk1.4.2\bin (jos SDK on asennettu C:\j2sdk1.4.2 hakemistoon), samalla voit lisätä CLASSPATH muuttujan ja sen arvoksi:
.;c:\mhp\mhp;c:\mhp\Javatv_fcs\Javatv.jar (jos MHP on asennettu c:\mhp\...)
- Lisää kansio c:\mhp\mhp\projects (jos MHP on asennettu c:\mhp\...)
- Asenna ImageMagick-kuvankäsittelyohjelma, jonka avulla voidaan näyttää täysivärisiä taustakuvia vaikka MPEG2-Iframea ei olisikaan. Ohjelma yrittää muuttaa MPEG2-Iframe JPEG-muotoon ja yrittää sitten näyttää JPEG-päätteisen kuvan, jos ohjelmaa ei löydy taustakuvana näkyy oletuskuva.

OpenMHP käynnistyy kaksoisklikkaamalla start.bat tiedostoa. Sovellus, jota halutaan testata, voidaan lisätä Application Manageriin painamalla hiiren oikeaa nappia ja valitsemalla add. Anna projektille nimi, jolla haluat sen näkyvän ohjelmalistassa. Valitse tiedostopolku josta ajettava sovellus löytyy, sekä käännetty tiedosto josta ajettava Xlet löytyy. OrgID ja AppID kohdilla ei ole tässä tapauksessa merkitystä. [15]

5.2 OpenMHP

OpenMHP on täysin ilmainen vapaaseen levitykseen ja käyttöön tarkoitettu digitaalisen television MHP-standardia noudattava avoin ohjelmistokirjasto lähdekoodeineen. Kirjasto sisältää kaksi osaa Adaptation Layer- ja MHP-kirjaston. MHP-kirjasto sisältää ne MHP-määrittelyn osat, jotka ovat samoja kaikille alustoille. Adaptation Layer sisältää ne osat MHP-määrittelyä, joita tarvitaan tietyillä

alustoilla (esim. sulautetut järjestelmät). OpenMHP-ympäristö ei tarjoa vielä kaikkia MHP-standardin mukaisia palveluita käyttöön, mutta sen kehitys jatkuu koko ajan.

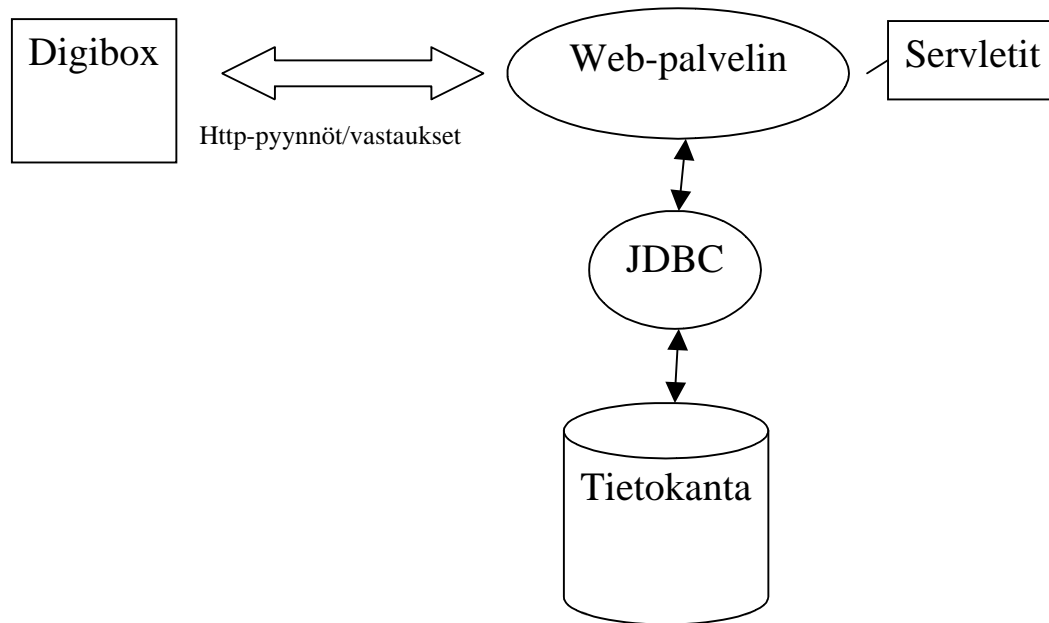


Kuva 11 OpenMHP:n emulaattori ja äänestys-sovellus.

OpenMHP sisältää emulaattorin, jolla voidaan testata luotuja digi-tv-sovelluksia ja niiden toimintaa. Emulaattori sisältää kaukosäätimen ja tv-ruudun (Kuva 11), joiden avulla voidaan testata sovelluksen toimintoja. Lisäksi emulaattorissa on tulostusruutu, jonka avulla voidaan seurata sovelluksen kulkua ja tapahtumia. [15, 21]

5.3 Asiakas-palvelin -tiedonsiirto

Kuvassa 12 on kuvattu karkealla tasolla asiakas-palvelin ohjelmiston tiedonsiirron kulku. Digiboxi ja web-palvelin keskustelevat toistensa kanssa Http-protokollan välityksellä. Palvelimen toiminnallisuus on tehty Java servletteinä ja JDBC-sillan avulla palvelin kykenee kommunikointiin tietokannan kanssa.



Kuva 12. Tiedonsiirron kulku

5.4 Järjestelmän toiminnot

5.4.1 Perustila

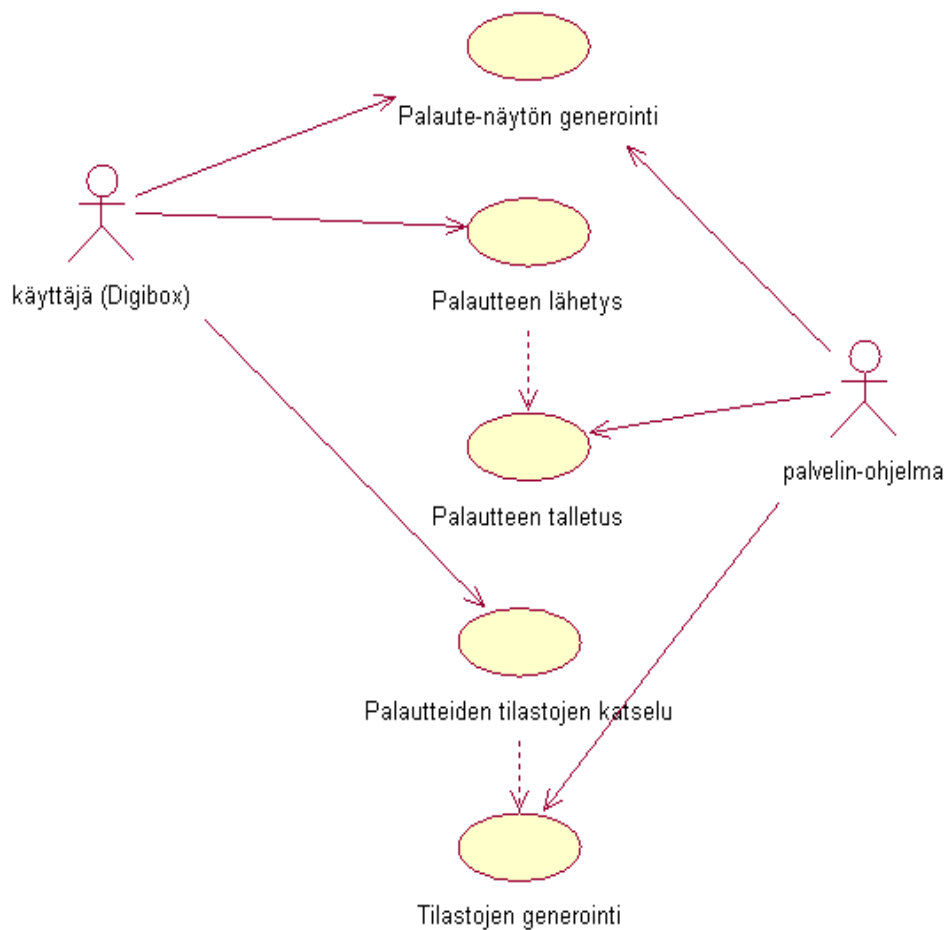
Sovellus muodostaa alkuvalikon, josta käyttäjä pääsee äänestyskseen. Ohjelma jää odottamaan käyttäjän valintaa.

5.4.2 Äänestys

Sovellus lähettää palvelimelle HTTP POST-pyynnön mukana ohjelman yksilöidyn tunnuksen, jonka perusteella palvelin hakee tietokannasta ohjelmakohtaiset tiedot ja luo merkkijonon, jonka se lähettää asiakkaalle. Palvelimelta saadun merkkijonon perusteella muodostetaan äänestysvalikko, josta käyttäjä voi valita äänestysvaihtoehdon painamalla OK nappia halutun äänestysvaihtoehdon kohdalla. Sovellus lähettää annetun äänen, ohjelman ja käyttäjän tiedot eteenpäin palvelimelle. Järjestelmä ilmoittaa, mikäli ääni on annettu onnistuneesti.

5.4.3 Annettujen äänien tulostus

Sovellukseen oli tarkoitus sisällyttää annettujen äänien tarkastelu mahdollisuus, mutta toistaiseksi sovellus ilmoittaa vain toiminnon puutteesta. Tulevaisuudessa käyttäjän on mahdollista valita valikosta annettujen äänien näyttäminen, palvelin laskee tietokannasta annetut äänet ja lähettää määrät merkkijonona digiboxille, joka muodostaa tuloksista graafisen esityksen näytölle.



Kuva 13. Kuvaus järjestelmän toiminnoista.

Asiakasohjelman käyttäjä (kuvassa 13 käyttäjä (Digibox)) antaa kaikki toiminnot järjestelmälle oman kaukosäätimen kautta. Jokaiselle ohjelmalle annettu palaute välittyy palvelimelle sillä hetkellä, kun kaukosäätimestä painetaan OK näppäintä. Palvelin lähettää varmistuksen käyttäjälle, kun ääni on tallennettu.

5.5 Ohjelmistoarkkitehtuuri, moduulit ja prosessit

Moduulin nimi: Vote

Pääluokka joka määrittelee piirtopinnan ja kuuntelee kaukosäätimen näppäimiä.

Pääluokka joka käynnistetään ensimmäisenä. Xletin luonti funktiot.

Luokka kontrolloi Xletin käynnistämistä ja pysäyttämistä. Piirtää ruudulle grafiikan.

Rajapintafunktiot:

InitXlet

Parametriksi ottaa XletContext tyyppisen ctx muuttujan. Ei palauta mitään.

Funktio määrittää piirtopinnan jolle voidaan mm. äänestysvalikko piirtää. Lisää piirtopinnalle halutut komponentit.

StartXlet

Ei tarvitse parametrikksi mitään. Ei palauta mitään.
Käynnistää Xletin mikäli se on pysäytetty.

PauseXlet

Ei tarvitse parametrikksi mitään. Ei palauta mitään.
Pysäyttää Xletin ja jättää sen ”taustalle” päälle.

DestroyXlet

Ottaa parametrikksi boolean arvon b. Ei palauta mitään.
Jos Xlet on käynnissä, tämä tuhoaa sen.

Moduulin nimi: GUI

Luokassa luodaan äänestysvalikon annettujen parametrien perusteella. Otsikkotiedot äänestysvalikolle tulevat myös palvelimelta. Parametreina annetaan: (string text). Stringi on esimerkiksi muotoa: 1|0|4|Idols äänestys|3|0|5|Antti|3|0|6|Seppo|3|0|7|Jouko. Luokka muodostaa ykkösen jälkeisestä tekstistä äänestysvalikolle otsikon ja niistä jotka tulevat kolmosten jälkeen, valikon jossa nimet ovat kirjoitettuna. Nollia ei tarvitse Idols äänestys -mallin mukaisessa valikossa ottamaan huomioon. Nollan tilalle tulee arvosanaäänestyksen kohdalla luku joka vastaa sitä, kuinka monen numeron skaalalla arvostelu annetaan. Idols äänestys -mallin mukaisesti voidaan luoda myös Kyllä / Ei äänestys. Kolmas numero (4,5,6,7 esimerkissä) kertoo palvelimelta tulevan kysymys id:n. Tämä id tieto lähetetään palvelimelle siinä yhteydessä kun käyttäjä antaa äänen.

Rajapintafunktiot:

Click

Funktio joka kutsutaan kun käyttäjä antaa äänensä. Syöttää parametrillisen kutsun Http_con luokan SendVote funktiolle. Ei palauta mitään

getButtons, getBut_st, getHeader, getVoteID

Funktioita joilla saadaan erinäisiä tietoja poimittua stringistä jonka perusteella äänestysvalikko luodaan. Tärkeimpänä getVoteID jota tarvitaan kun lähetetään palvelimelle tiedot annetusta äänestä.

Moduulin nimi: http_con

Luokka joka huolehtii server -ohjelmiston ja client -ohjelmiston tiedonsiirrosta
Luokka lähettää palvelimelle tietoa ja vastaanottaa sitä.

Rajapintafunktiot:

GetMenu

Funktio ottaa parametrikksi ohjelmaID:n. Palauttaa palvelimelta saadun vastauksen.
Funktio lähettää parametreina saadut tiedot servletille ja vastaanottaa sieltä vastauksen.
Jotta funktio voidaan suorittaa, tulee olla ohjelmavirrasta haettu ohjelmaID.

SendVote

Funktio ottaa parametrikksi OhjelmaID:n, digiboxID:n sekä käyttäjän antaman äänen.
Palauttaa kuittauksen palvelimelta.
Funktio lähettää parametrit servletille ja vastaanottaa sieltä kuittauksen lähetykseen.
Funktio suoritetaan, kun GetMenu on suoritettu ja käyttäjä on antanut äänensä.

GetStatistics

Funktio ottaa parametriksi OhjelmaID:n ja palauttaa äänestystilastot.
Funktio lähettää parametrin servletille ja vastaanottaa sieltä äänestystiedot kyseisestä ohjelmasta. Ei toistaiseksi käytössä

Moduulin nimi: RetrieveProgID

Luokka joka hakee ohjelmavirrasta ohjelmaID:n.

Rajapintafunktiot:

GetProgID

Funktio ei ota parametreiksi mitään. Palauttaa ohjelmavirrasta haetut ohjelmaID:n osat.
Funktion voi kutsua milloin vain, välittämättä siitä missä tilassa ohjelma kulloinkin on.

Moduulin nimi: FinishMenu

Luokka jossa piirretään valikko joka ilmaantuu äänen antamisen jälkeen.
Luokalla ei ole rajapintafunktioita

Moduulin nimi: TextBox

Luokka jossa piirretään tekstiruutu jolle voidaan tulostaa haluttuja tietoja.
Luokalla ei ole rajapintafunktioita.

Moduulin nimi: VideoHandler

Luokka jossa käsitellään tv-kuvaa. Jotta sovelluksen taustakuva näkyisi, täytyy videotaso piilottaa näkyvistä.
Tärkein funktio setBg-funktio, joka saa parametreinä pisteet johon uusi videotaso sijoitetaan ja arvot joilla nykyistä kuvaa skaalataan.

Rajapintafunktiot:

show

Funktio kutsuu setBg-funktiota, joka asettaa videotason koko tv-ruudun kokoiseksi.

hide

Funktio kutsuu setBg-funktiota, joka asettaa tv-kuvan skaalattuna oikeaan alakulmaan.

5.6 Virheidenkäsittely

Sovelluksessa ei ole ajanpuutteen vuoksi toteutettu virheenkäsittelyä. Järjestelmä ei toivu virheistä ja vain joistain toteuttamattomista toiminnoista järjestelmä antaa ilmoituksen.

5.7 Testaus

Itse kehitetyt sovellukset vaativat paljon testausta. Eri vastaanottimet toimivat eri tavoin ja toteuttavat eri ominaisuuksia MHP-määrittelystä. Koska vastaanottimet ja emulaattorit eivät toimi samalla tavoin, vaativat muutokset testausta myös loppulaitteessa. Testaus vastaanottimessa vaatii kuitenkin huomattavasti enemmän resursseja mm. laitteet ja aikaa, kuin testaus emulaattorissa.

Jotta toimintojen oikeellisuudesta voitaisiin varmistua, sovelluksen kaikki ominaisuudet tulisi testata kaikilla markkinoilta löytyvillä vastaanottimilla. Lisäksi testaus tulisi toistaa aina kun johonkin laitteeseen tulee saataville ohjelmistopäivitys.

Tässä projektissa jokainen koodaaja on suorittanut lasilaatikkotestausta omaa moduuliaan tehdessään. Koska projektin käytössä oli yksi yhteinen digiboxi, voitiin loppulaitetestausta suorittaa vain sovittuina ajankohtina.

Projektin päävaiheet jakautuivat lähetettävien ohjelmatietojen hankkimiseen ja käsittelyyn, sekä graafisten komponenttien toteutukseen. Ohjelmatietoja käyttävien moduulien testaus jouduttiin toteuttamaan aidossa ympäristössä, kuten myös ohjelma virtaa käyttävien graafisten komponenttien testaus. Tämä hankaloitti sovelluskehitystä huomattavasti, mm. testaamisen hitauden takia.

Koska kyseessä on asiakas-palvelin tyyppinen järjestelmä on asiakas-puolen moduulit yhdistetty, jonka jälkeen asiakas-sovelluksen ja palvelimen välistä toimintaa on testattu ns. harmaalaatikkotestauksella. Lopuksi järjestelmälle suoritettiin järjestelmätestaus. Ajanpuutteen vuoksi järjestelmästä löytyneitä virheitä ja puutteita ei pystytty korjaamaan, vaan ne dokumentoitiin jatkokehittäjien käyttöön.

6 Yhteenveto

DIGITAALISEN TELEVISION JA MHP-STANDARDIN VAHVUUDET

Digitaalisen television väestöpeitto on Suomessa jo yli 70%. Päätelaitteiden valmistajien ja mallien lisääntyminen markkinoilla aiheuttaa myös hintojen laskua, mikä madaltaa kuluttajien kynnystä hankkia digiboxi. Käyttäjien näkökulmasta digitaalisen television toistaiseksi tärkeimmät ominaisuudet ovat kuvan ja äänen parempi laatu, sekä supertekstitelevisio ja kanavien lisääntyminen. Vuorovaikutteiset palvelu- ja hyötysovellusten, kuten pankki- ja kauppapalvelut, lisääntyminen helpottaa käyttäjien jokapäiväistä elämää. Tarpeettomien sovellusten lataaminen on helppo estää, koska käyttäjä voi itse valita mitä sovelluksia digiboxille ladataan. Avoin ja laajalti leviävä MHP-standardi takaa markkinoiden laajuuden. Sovelluksia voi tehdä kuka tahansa, esimerkiksi tässä työssä käytetyssä OpenMHP ympäristössä. Perus Java-kielen taidoilla, sekä tutustumisella käytössä oleviin paketteihin onnistuu yksinkertaisen sovelluksen teko. Avoin standardi lisää tekijöiden määrää ja nopeuttaa standardin kehitystä. Uusia sovelluksia syntyy jatkuvasti nopeammalla tahdilla.

DIGITAALISEN TELEVISION JA MHP-STANDARDIN HEIKKOUEDET

Toistaiseksi markkinoilla olleiden digisovittimien valikoima ja yhteensopivuus on ollut heikko. Kaapeliverkkoon tarkoitettuja päätelaitteita ei ole ollut tarjolla juuri lainkaan. Laitteiden hidas toiminta esimerkiksi parin sekunnin viive kanavaa vaihtaessa tai sovelluksia ladatessa, saattaa tuntua oudolta analogiseen televisioon tottuneelle. Koska MHP-standardi on varsin nuori standardi, on siinä edelleen puutteita ja kehitettävää, näistä ongelmista kärsii eniten edelläkävijämaat, joihin Suomikin kuuluu. Tässä työssä käytetty OpenMHP ympäristö osoittautui vielä melko keskeneräiseksi. Puuttavia ominaisuuksia oli mm. mahdollisuus ääni- ja kuvavirtojen käyttöön, sekä useat toteuttamattomat MHP-luokat. Useat sovelluksen ominaisuudet jouduttiin testaamaan emulaattorin sijaan oikeassa ympäristössä.

DIGITAALISEN TELEVISION JA MHP-STANDARDIN MAHDOLLISUUDET

MHP-standardi on levinnyt jo useisiin euroopan suurimaihin, jos kehitys jatkuu samansuuntaisena, standardin kehitys nopeutuu sekä palveluntarjoajien ja päätelaitteiden määrä että kilpailu lisääntyy. MHP-sovelluskehitykseen tarkoitettuja sovelluksia on markkinoilla jo useita, niiden edelleen kehittyessä ja lisääntyessä helpottuu sovellusten teko ja tekijäkunta kasvaa. Tulevaisuudessa suuri osa mm. internetissä olevista palveluista voidaan siirtää käytettäväksi digiboxeilla. DVB-H-standardin tuomat aika- ja paikkariippumattomat palvelut mobiiliverkoissa tuovat mukanaan jälleen uusia mahdollisuuksia joidenka kehittämisessä ja toteuttamisessa Suomi on edelläkävijä.

DIGITAALISEN TELEVISION JA MHP-STANDARDIN UHAT

Jos MHP-standardin leviäminen uusiin maihin pysähtyy, jää käyttäjäkunta pieneksi ja standardin kehitys hidastuu. Panostus verrattuna muihin kilpaileviin standardeihin jää pieneksi ja laitteiden ja palveluiden tarjoajat vähentyvät. Mikäli digiboxien toimivuus ja valikoima eivät lisäänty tarpeeksi nopeasti, käyttäjät lykkäävät digiboxien hankintaa, josta seuraa palveluiden hidas lisääntyminen, koska palveluille ei ole käyttäjiä. Uusien tietokoneen ja television ominaisuudet sisältävien multimediasovelluksien lisääntyminen saattaa vähentää tarvetta digiboxin hankinnalle ja uusille MHP-sovelluksille. Sovellusten toiminta tulee pitää selkeänä ja helppokäyttöisenä, jotta kynnys niiden käyttöön kaukosäätimellä ei kasva liian suureksi verrattuna tietokoneeseen.

TYÖN VAIHEET

Työssä keskityttiin MHP-standardin mukaiseen sovelluskehitykseen OpenMHP-ympäristössä. Tutkintotyön ohessa syntyi LähiTV-hankkeen käyttöön toimiva äänestys-sovellus, jonka kehittämistä jatketaan Tampereen ammattikorkeakoulussa. Insinöörityön tarkoitus on myös helpottaa sovelluksen jatkokehittäjien tutustumista MHP-sovelluskehitykseen.

Suurin osa projektin suunnitelluista ominaisuuksista saatiin toteutettua. Lähes kaikki ajanpuutteen vuoksi toteuttamatta jääneet toiminnot ehdittiin myös suunnitella valmiiksi. Suurin osa työhön käytetystä ajasta kului perustietojen hankintaan, koska kenelläkään projektiin osallistuneista ei ollut aikaisempaa kokemusta MHP-sovelluksista. Alussa hyödyllisimmiksi tietolähteiksi osoittautuivat Arvidin tekemä Digi-tv:n palveluntekijän opas (lähde 15), sekä sivustoilta löytyvät esimerkkiohjelmat, lisäksi perustietoa digitaalisesta televisiosta löytyi seuraamalla Digi-TV sovelluskehittäjän linkkilista -sivuilta löytyviä linkkejä (lähde 23). Näiden avulla kehitysympäristö saatiin toimintaan ja mm. käyttöliittymän suunnittelussa päästiin alkuun. Jatkossa todella hyödyllisiksi sivustoiksi osoittautuivat Interactive TV Web (lähde 18), josta löytyy sekä opas että tarvittavat tiedot ja hyödylliset esimerkit MHP-sovellusten tekoon. Hyödyllistä lisätietoa löytyi myös OpenMHP ja Java -foorumeilta (lähteet 21 ja 22). OpenMHP-foorumi vaatii rekisteröitymisen.

Työn päävaiheet jakautuivat lähetettävien ohjelmatietojen hankkimiseen ja käsittelyyn, sekä graafisten komponenttien toteutukseen. Koska lähetettävistä ohjelmista on tarjolla useita erilaisia informaatioita, olisi sovelluksessa voitu käyttää myös erilaista ohjelmatietojen käsittelytekniikkaa. Ehdotukset mahdollisista muutoksista toteutukseen dokumentoitiin ja jätettiin tulevalle jatkokehitysryhmälle.

Koko projektiin osallistunut ryhmä koki projektin mielenkiintoisena, minkä takia työ eteni aikataulussa ja työmäärät jakautuivat tasaisesti osallistujien kesken. Projektiryhmä oli tyytyväinen lopputulokseen ja projektin aikana tapahtuneeseen oppimiseen.

LÄHTEET

- 1 pro.sony.com.hk/pr20010914h.html
- 2 http://homeplatform.broadcastengineering.com/yle_launches_dvb_mhp/
- 3 <http://www.minedu.fi/julkaisut/kulttuuri/2004/tr19/tr19.pdf>
- 4 http://www.mhp.org/news_and_events/news/archive/
- 5 www.media.hut.fi/~julkaisut/diplomityot/DI_J_Haataja.pdf
- 6 www.mintc.fi/www/sivut/dokumentit/julkaisu/julkaisusarja/2002/a022002.pdf
- 7 <http://www.ficora.fi/suomi/radio/digitv.htm>
- 8 http://www.tekes.fi/julkaisut/Digi_TV_opas.pdf
- 9 www.media.hut.fi/~julkaisut/diplomityot/DI_R_Hiidenkari.pdf
- 10 http://www.mhp.org/about_mhp
- 11 <http://www.bbc.co.uk/rd/pubs/presentations/pdf/ibc00jcn-ppt.pdf>
- 12 http://www.etsi.org/SERVICES_PRODUCTS/ptcc/home.htm#
- 13 www.media.hut.fi/~julkaisut/diplomityot/DI_T_Venho_2004.pdf
- 14 pww.evitech.fi/courses/mm2002/insinoorityot/vmp99s/kyllastinen/DigitaalisenTelevisionKokeiluymparisto.pdf
- 15 www.arvid.tv Digi-tv:n palveluntekijan opas
- 16 Digitaali TV-tekniikka, Olli Ojala. Opetusmateriaali
- 17 <http://www.yle.fi/tekniikka/tklehti/tk81/karu.htm>
- 18 <http://www.interactivetvweb.org/tutorial/mhp/index.shtml>
- 19 www.saunalahti.fi/osavia/diplomityo/diplomityo.pdf
- 20 <http://java.sun.com/developer/technicalArticles/javatv/apiintro/>
- 21 <http://www.openmhp.org/>
- 22 <http://forums.java.sun.com/forum.jspa?forumID=36>
- 23 <http://koti.mbnet.fi/digitale/mhp/mhp.html>